



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Modelo y lecciones aprendidas del proceso de creación de MOOCs para enseñar a programar

Citation for published version:

Friss De Kereki, I & Manataki, A 2016, Modelo y lecciones aprendidas del proceso de creación de MOOCs para enseñar a programar. in *Sexta Conferencia de Directores de Tecnología de Información, TICAL 2016 Gestión de las TICs para la Investigación y la Colaboración*. pp. 171-184, Conference of the Network of Directors of the Information and Communication Technologies in Latin American Universities, Buenos Aires, Argentina, 13/09/16. <<http://tical2016.redclara.net/index.php/en/organizacion/sobre-tical/actas-de-las-conferencias-anteriores>>

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Sexta Conferencia de Directores de Tecnología de Información, TICAL 2016 Gestión de las TICs para la Investigación y la Colaboración

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Modelo y lecciones aprendidas del proceso de creación de MOOCs para enseñar a programar

Inés Friss de Kereki^a, Areti Manataki^b

^a Facultad de Ingeniería, Universidad ORT Uruguay, Cuareim 1451,
11100, Montevideo, Uruguay
kereki_i@ort.edu.uy

^b School of Informatics, University of Edinburgh, 10 Crichton Street,
EH8 9AB, Edinburgh, United Kingdom
A.Manataki@ed.ac.uk

Resumen. El pensamiento crítico y la programación de computadoras promueven el desarrollo de habilidades cognitivas cada vez más necesarias para los jóvenes. Los MOOCs (cursos masivos abiertos en línea) son herramientas valiosas para popularizar el acceso a la educación. La Universidad ORT Uruguay ha desarrollado en forma individual y también en conjunto con The University of Edinburgh varios MOOCs orientados a fomentar en los jóvenes el acercamiento al pensamiento computacional y al aprendizaje de la programación. Los resultados de las encuestas de los MOOCs realizados hasta el momento son muy alentadores y reflejan un alto grado de conformidad con los cursos. En este trabajo se presentan las características de estos MOOCs y sus resultados y se brinda un posible modelo del proceso de creación de MOOCs así como las lecciones aprendidas. En dicho modelo se describen las principales etapas, tareas y roles vinculados. Esta propuesta, orientada fundamentalmente a MOOCs para enseñanza de la programación, podría ser además extrapolable a otras áreas.

Palabras Clave: MOOC, Programación, Enseñanza

1 Introducción

Para interactuar en la sociedad digital, todo ciudadano en el presente debe al menos entender los principios de las ciencias de la computación [1]. Las habilidades que se obtienen al codificar ayudan a interpretar la sociedad actual y fomentan las competencias del siglo 21 [2]. Así, hoy en día, cada vez más en los planes de estudio curriculares, tanto de enseñanza primaria como de enseñanza secundaria, se incluye el estudio formal de las habilidades y competencias computacionales [1][3].

Los MOOCs (cursos masivos abiertos en línea, por sus siglas en inglés: “Massive Open Online Course”) son una poderosa herramienta para popularizar el acceso a la educación y democratizar el conocimiento. Tienen el potencial de hacer accesibles las ofertas educativas a nivel mundial [4] y permiten a los estudiantes acceder a contenidos académicos de alta calidad [5].

Dada la importancia creciente de conocer y adquirir dichas habilidades relativas a ciencias de la computación y también con los objetivos de mayor inclusión e igualdad de oportunidades, desde 2013 la Universidad ORT Uruguay en forma individual y también en forma conjunta con The University of Edinburgh, ha desarrollado varios

MOOCs orientados a la enseñanza inicial de la programación, dirigidos fundamentalmente a jóvenes. En este trabajo se describe en primera instancia qué son los MOOCs y se analizan diversas recomendaciones aplicables al proceso de su creación. Luego, se detallan en particular los cuatro MOOCs realizados por estas universidades para aprender a programar. A partir de estas experiencias, se propone un modelo para el proceso de creación de un MOOC y se ofrecen lecciones aprendidas. Finalmente se brindan conclusiones.

2 MOOCs

Un MOOC es un curso en línea, masivo y abierto, ofrecido a través de Internet y disponible de forma gratuita a un número muy grande de personas [6][7]. Los MOOCs promueven el desarrollo de las habilidades de colaboración de los individuos con otras personas en línea, para participar en la economía digital. [8].

Un MOOC está típicamente formado por videos, textos, imágenes y evaluaciones. Además, agrega foros de discusión e intercambio para los participantes [9].

Hay múltiples recomendaciones en cuanto al proceso de creación de un MOOC, el cual implica fases de diseño, desarrollo y producción. En [10] se sugiere que para crear un MOOC es útil participar previamente como estudiante en varios MOOCs para obtener inspiración e ideas. En cuanto a la temática específica del MOOC, se indica allí [10] elegir temas que despierten interés en los estudiantes. En relación al proceso, se aconseja armar un equipo, planificar el desarrollo, crear las evaluaciones y tener en cuenta también la tecnología disponible así como definir la duración del curso, cuya recomendación es que sea entre 4 y 8 semanas [10].

Para diseñar un MOOC exitoso, Yousef et al [11] refieren a considerar criterios específicos relacionados tanto a pedagogía (diseño instruccional y evaluación) como a aspectos que denominan técnicos (interfaz de usuario, contenido, aprendizaje y herramientas sociales entre otros). En particular, en relación al diseño instruccional destacan como más importante la organización de la lección.

The University of Edinburgh ofrece un posible modelo para el desarrollo [9]. Dicho modelo incluye recomendaciones generales para el proceso de creación de un MOOC específicamente en Coursera, una de las plataformas más reconocidas [12] [13]. Entre ellas podemos citar revisar ejemplos de lecciones en video y tener en cuenta la elección de imágenes de uso libre. Esta guía puede ser de orientación en las primeras etapas del proceso.

Spyropoulou et al [14] describen como metodología estas cinco fases: 1. Fase de Análisis: se analiza el tópico del curso, 2. Fase de Diseño: se diseñan los objetivos de aprendizaje y actividades asociadas, 3. Fase de Desarrollo: se desarrollan los diversos elementos constitutivos del curso, 4. Fase de Implementación: el proceso educacional se implementa y se prueba el curso en forma “piloto”, y 5. Fase de Evaluación: se evalúan todas las fases. Es particularmente interesante la conformación de los equipos de trabajo, que incluyen: a) equipo académico: docentes y docentes asistentes para el diseño del curso e implementación, b) equipo técnico, refiere a personal técnico para la operación, mantenimiento y operación de la plataforma del curso, c) equipo de

soporte en educación y d) equipo adicional, por ejemplo, responsable de desarrollos multimedia.

Collazos [15] propone un modelo de ocho pasos para la creación de un MOOC: 1. Planeación del curso donde se indican objetivos genéricos, 2. Estructuración del curso: división en unidades y temas, 3. Selección de un tema particular, para tomar como “piloto”, 4. Realización de un guión gráfico, con viñetas para saber qué grabar y cómo, 5. Filmación, recomienda comenzar con grabaciones caseras, 6. Repetir los pasos 4 y 5 para los demás videos y 7. Subir los videos a Internet. Finalmente indica como etapa 8 y final la promoción del curso. Estas etapas son generales y parecen simplificar demasiado el proceso de crear un curso en línea masivo de calidad ya que no incluyen, por ejemplo, recomendaciones sobre el diseño de los videos o la evaluación.

Alario et al [16] refieren también a verificar qué elementos están disponibles en la plataforma que se use para diseñar el contenido y las actividades de evaluación así como dejar tiempo suficiente para poder probar todos los materiales antes de ponerlo disponible a los estudiantes.

Morrison brinda recomendaciones específicas sobre elementos particulares de un MOOC [17]: sugiere incluir una página de comienzo como orientación a los estudiantes, que las actividades impliquen que los estudiantes “hagan” tarea y que las evaluaciones sean nuevas oportunidades para aprender. Bali [18] recomienda fomentar el aprendizaje activo y dar retroalimentación de los cuestionarios lo antes posible. Sugiere considerar el balance entre lo requerido para completar el curso con el aprendizaje crítico y profundo. La duración sugerida es de cuatro semanas [17] [19].

En relación a los videos, elemento central de los MOOCs, Guo et al [20] refieren que la mayoría de los videos en MOOCs son lecciones (el profesor explica un cierto tema) y tutoriales o demostraciones (por ejemplo se explica paso a paso un proceso). En cuanto a los estilos de producción, citan varios como el formato de profesor hablando (con o sin imágenes de Power Point), el profesor grabado durante una clase real y el profesor grabado en estudio de grabación, sin alumnos, entre otros. Para elegir cuál conviene, hay que tener en cuenta los objetivos y contenidos del MOOC. También recomiendan usar videos cortos ya que son mucho más “atrapantes”, así como los videos donde los instructores muestran gran entusiasmo y hablan rápido, lo que se logra a través del entrenamiento. No es necesario incluir pausas en los videos ya que el estudiante puede detener y volver a ver las partes que desee [20 GUO]. Recomendán invertir en la pre-producción de los videos, para planificar las lecciones específicamente para ese formato.

Hibbert [21] señala que el tiempo promedio de visualización de un video por parte de un estudiante es de cuatro minutos. También señala que la experiencia del usuario viendo videos de baja calidad de imagen o sonido no es buena. Button [19] recomienda, además de que los videos sean de pocos minutos, que los profesores proyecten un aire más “informal” sentados frente a un escritorio que de pie en un podio [19]. Alario et al [16] también señalan entre 5 y 10 minutos como la duración recomendada de los videos y combinar diversos tipos de recursos en los mismos, tales como animaciones, dibujos o capturas de pantalla. Aspectos similares en cuanto a duración de videos (entre 3 y 5 minutos) así como hacer atractivos los videos usando animaciones, links y textos son citados en [22].

De acuerdo con Adamopoulos [23], cuanto más satisfecho esté el estudiante con el profesor, los materiales y las tareas a realizar, más probablemente termine el curso exitosamente.

3 MOOCs de la Universidad ORT Uruguay y The University of Edinburgh para aprender a programar

Desde 2013 y a mayo de 2016, la Universidad ORT Uruguay creó varios MOOCs para aprender a programar según se presentan en la Tabla 1. Podemos citar: a) MOOC “SM4T” (“Scratch MOOC for teens”) realizado en 2013 [24] para Plan Ceibal [25], b) MOOC “¡A Programar!” [26] y “Code Yourself!” [27], diseñado conjuntamente con The University of Edinburgh, disponibles desde marzo de 2015. También se incluyen los MOOCs de próximo lanzamiento, realizados por la Universidad ORT Uruguay para Plan Ceibal: c) “Scratch 2.0 Avanzado” y d) “App Inventor”. Los tres primeros utilizan como lenguaje Scratch [28] y el último utiliza MIT App Inventor [29]. Estos lenguajes fueron elegidos por múltiples razones. Entre ellas, por ser lenguajes y ambientes gratuitos, disponibles online, estar orientados a jóvenes (se pueden crear animaciones y juegos muy fácilmente), ser visuales, esto es, se arrastran y encastran bloques para programar, así como la amplia experiencia de los docentes a cargo en estas herramientas. Las plataformas utilizadas, en el caso de CEIBAL son CREA y CREA2 (basadas respectivamente en las plataformas Educativa [30] y Schoology [31]) y Coursera [12], para el MOOC conjunto con The University of Edinburgh.

Tabla 1. MOOCs para aprender a programar

MOOC	Fecha lanzamiento	Plataforma	Alcance	Cantidad de participantes	Conformidad con el curso (encuestas)
SM4T: “Scratch MOOC for teens”	Octubre 2013	CREA (CEIBAL)	Nacional (Uruguay)	+1.190 en la primera sesión	97% “muy bueno” o “bueno”
¡A Programar!/ Code Yourself!	Marzo 2015	Coursera	Internacional (más de 117 y 197 países respectivamente)	+140.000 acumulado desde 3/2015	95% “colmó o superó expectativas”
Scratch 2.0 Avanzado	Mayo 2016	CREA2 (CEIBAL)	Nacional (Uruguay)	+2200 (pre-inscripciones a 5/2016)	(no disponible aún)
App Inventor	Junio 2016	CREA2 (CEIBAL)	Nacional (Uruguay)	No abierto aún	(no disponible aún)

Describiremos brevemente cada uno de ellos. Los cursos son:

SM4T (Scratch MOOC for Teens: MOOC de VideoJuegos y Animaciones en Scratch) [24]. Este MOOC está dirigido a jóvenes de 12 a 17 años, sin conocimientos de programación y dura cinco semanas, siendo su alcance nacional. Sus objetivos son promover el interés por las Tecnologías de la Información y de la Comunicación (TICs), promover habilidades para resolución de problemas, fomentar el pensamiento lógico y la creatividad. Al final del curso los estudiantes serán capaces de: a) identificar y usar apropiadamente los componentes básicos de Scratch 1.4: bloques, variables, escenarios, “sprites”, imágenes, sonidos y ejecutar el programa, b) construir programas sencillos a través del uso de variables simples, estructuras de control y manejo de eventos, combinando los diferentes componentes de la herramienta, c) comparar, evaluar y revisar programas sencillos utilizando su propio juicio crítico.

En la primera edición participaron más de 1190 personas. 97% de quienes respondieron las encuestas consideraron el curso como muy bueno o bueno. Las primeras ediciones estuvieron disponibles en la plataforma CREA de CEIBAL y actualmente está disponible en CREA2. En la Fig. 1. se adjunta una imagen tomada del curso.

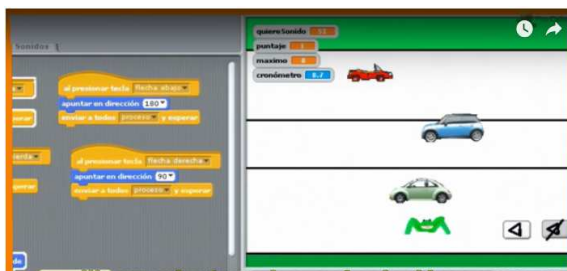


Fig. 1. Imagen de un video de “SM4T”

“¡A Programar!”/“Code Yourself!”. Este MOOC fue diseñado en forma conjunta por Universidad ORT Uruguay y The University of Edinburgh. Dura cinco semanas y está orientado a jóvenes sin conocimientos de programación. Los objetivos son fomentar el pensamiento computacional, conocer las prácticas básicas de la ingeniería de software y desarrollar programas. Está disponible en dos versiones: la versión en español se denomina “¡A Programar!” [26] y la versión en inglés “Code Yourself!” [27]. El lenguaje utilizado es Scratch 2.0. Entre los temas se incluyen: a) definición de algoritmo, nociones sobre estructuras de control: secuencia, decisión e iteración, desarrollo incremental de programas y prueba simple, b) nociones sobre descomposición de problemas, programación orientada a eventos, diseño de interfaz, y detección de requerimientos para la construcción de software, c) presentación de prácticas de la ingeniería de software para probar y documentar los programas, y d) nociones de reutilización de código, generalización, modularidad y uso de parámetros.

En la siguiente figura (Fig. 2) se muestra una captura de un video del curso en español, en la que se ve a la docente junto a un personaje animado, que fue incluido en el curso para interactuar y formular preguntas. Todos los contenidos del MOOC fueron diseñados en forma conjunta a distancia (con una única instancia presencial). La filmación de videos fue realizada en cada idioma, en forma local.



Fig. 2. Imagen de un video de “¡A Programar!”

El curso está disponible desde marzo 2015. Actualmente se ofrece con comienzo en cada mes. Hasta el momento, participaron más de 140.000 personas de 117 países en “¡A Programar!” y de 197 países en “Code Yourself!”. En relación a la opinión de los participantes, los elementos que resultaron más valiosos fueron los videos y los cuestionarios. En las encuestas, 93% (en “Code Yourself!”) y 95% (en “¡A Programar!”) de quienes respondieron indicaron que el curso cumplió o superó sus expectativas. Además, 96% (“Code Yourself!”) y 98% (“¡A Programar!”) de los encuestados indicaron que recomendarían el curso. O sea, tanto en el caso de SM4T como en los de “¡A Programar!” y “Code Yourself!” las encuestas realizadas mostraron altos grados de conformidad con el respectivo curso. Es importante destacar que, en todos los cursos, la amplia mayoría de los participantes indicaron no tener conocimientos previos de programación.

Los últimos dos MOOCs incluidos en la Tabla 1 refieren a MOOCs desarrollados por la Universidad ORT Uruguay para Plan Ceibal, que se lanzarán en breve. Ellos son:

Scratch 2.0 Avanzado. Es un MOOC dirigido a jóvenes con los objetivos de continuar con el desarrollo del pensamiento procedural y lógico a través de la resolución de problemas de programación más avanzados utilizando Scratch 2.0 e incorporar los nuevos elementos de esta versión. Tiene como requisito haber completado el curso SM4T o tener conocimientos básicos de Scratch. Al finalizar este curso de cinco semanas el estudiante conocerá las características de la nueva interfaz de Scratch 2.0, podrá construir programas más avanzados incorporando, entre otros, el uso de listas y la definición de procedimientos; incluirá en los programas elementos nuevos de Scratch 2.0 relativos al diseño de imágenes, video, interacción con la cámara y sonidos, clonación y desarrollará videojuegos y animaciones integrando todos los conceptos vistos. Se observa en la Fig. 3 una imagen de un video del curso.

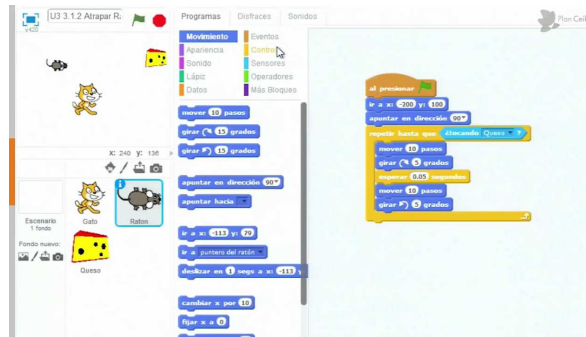


Fig. 3. Imagen de un video de “Scratch 2.0 Avanzado”

App Inventor. También es un curso dirigido a jóvenes, con una duración de cinco semanas. No tiene prerequisites. El propósito principal es despertar el interés por las TICs, promover el desarrollo de competencias emprendedoras, como la capacidad de innovación y desarrollo y la orientación al cliente, así como desarrollar las capacidades de pensamiento lógico en los estudiantes a través del desarrollo de aplicaciones para Android, utilizando la herramienta MIT App Inventor [29]. En el curso se crean varias aplicaciones (ejemplos: Aplicación para dibujar, Baliza luminosa, Block de Notas, Juego del Pong, ¿Dónde estoy?), a través de las que se presentan conceptos de estructuras de control, variable, lista y procedimiento, entre otros. En la Fig. 4 se muestra una imagen de uno de los videos del curso.

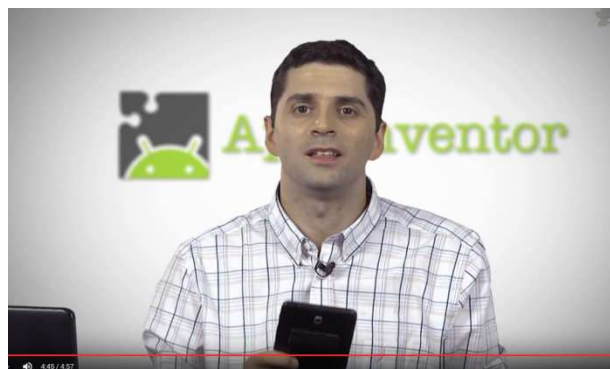


Fig. 4. Imagen de un video de “App Inventor”

4 Modelo propuesto y lecciones aprendidas

En función de las experiencias realizadas y con base en las distintas sugerencias y recomendaciones bibliográficas presentadas previamente, proponemos el siguiente modelo para el proceso de creación de un MOOC relativo al área de enseñanza de la programación. Este modelo ha sido aplicado y ajustado a través de los cuatro MOOCs

referidos, que incluyen trabajo de equipos localizados completamente en un mismo lugar y que hablan un mismo idioma, así como equipos en distintos continentes e idiomas diferentes (como en el caso de “¡A Programar!”/“Code Yourself!”). En cada paso del modelo se describen las características y las lecciones aprendidas, lo que puede ser de utilidad para la creación de futuros MOOCs orientados a esta misma área y quizás también extrapolables a otros contextos. Una característica particular a resaltar de este tipo de MOOC orientado a la enseñanza de la programación es que se integran imágenes de código de diferentes entornos de programación. O sea, es necesario incorporar en los videos permanentemente elementos de la interfaz gráfica del programa, lenguaje o ambiente que se esté usando (ej. Scratch) con el docente y los demás recursos.

Una vez definida la temática y el público objetivo, para producir un MOOC fundamentalmente hay dos equipos vinculados: uno docente y otro de producción, que trabajan conjuntamente y en coordinación. Los pasos sugeridos para crear un MOOC se muestran en la Fig.5. En la imagen, por cada paso se incluye el grado de participación de cada equipo en forma gráfica (rojo: docente, gris: producción).

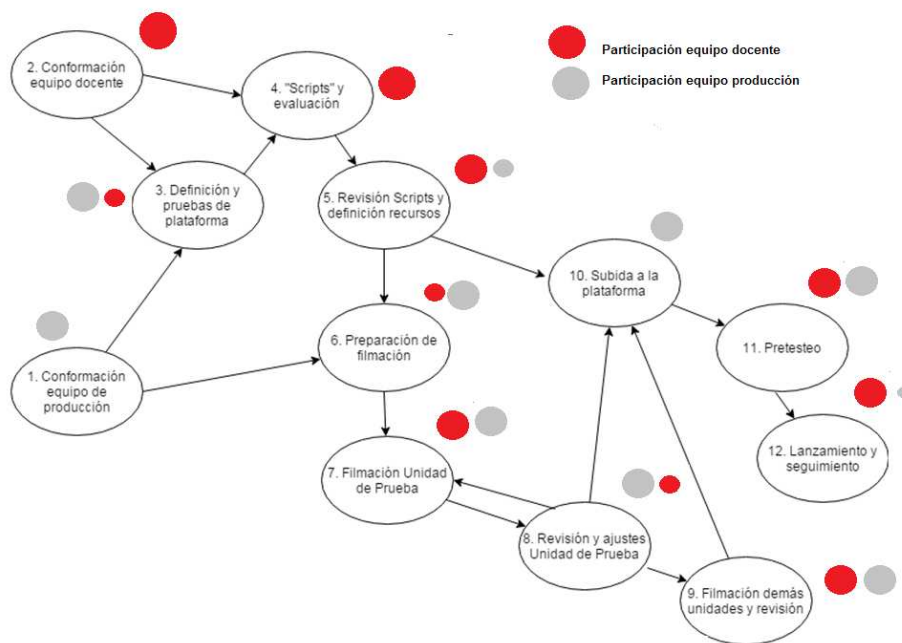


Fig. 5. Proceso sugerido de creación de un MOOC para aprender a programar

A continuación se detallan cada uno de los pasos. Estos son:

Conformación de equipo de producción (paso 1). En nuestro caso, el equipo está integrado por un coordinador general de todo el proyecto, un coordinador de producción y experto en plataformas, un productor audiovisual, un editor, un director de arte, un maquillador y, en caso de ser necesario, hay disponibilidad de diseñador

gráfico e ilustrador. Adicionalmente hay apoyo para la gestión de la plataforma y especialistas en educación y comunicación.

Conformación del equipo docente (paso 2). Hay un docente responsable de los contenidos temáticos del área del MOOC a construir y varios docentes que colaboran tanto en el desarrollo de los textos así como en la filmación, revisión y posterior seguimiento del curso. Como recomendación, sugerimos verificar previamente la disponibilidad horaria de los distintos docentes para todo el proceso, pues es una actividad altamente demandante. Por ejemplo, el curso “Scratch 2.0 Avanzado” llevó unos cinco meses entre escritura, filmación, revisión y subida a la plataforma, a dedicación horaria parcial de los docentes.

Definición y pruebas de plataforma (paso 3). La elección de la plataforma puede estar condicionada al propio proyecto en sí. En nuestro caso, cada proyecto ya tenía definida la plataforma a utilizar. Antes de comenzar a escribir los textos o durante este proceso es necesario hacer múltiples pruebas en la plataforma en la cual se subirá el curso para verificar efectivamente las opciones disponibles. Por ejemplo, las plataformas CREA y CREA2 de CEIBAL no permiten evaluación por pares por lo cual se debe tener en cuenta este punto para el diseño de la evaluación; la plataforma Coursera sí permitía evaluación por pares pero no dejaba subir archivos de tipo “.sb2” (requeridos para Scratch), se planteó el requerimiento a los técnicos de Coursera, quienes solucionaron este punto. Otro elemento adicional es la disponibilidad o no de los cuestionarios “en línea”, embebidos en los videos. Coursera permite su uso. También se debe analizar qué estilo de páginas y formatos permite la plataforma, por ejemplo, si permite definir por cada unidad una única página con recursos de distintos tipos (videos, archivos, links) o si se organizan por separado. Adicionalmente, otro elemento a definir es si se puede establecer o no precedencia obligatoria para acceder a los materiales. En esta etapa hay gran participación de integrantes del equipo de producción, interactuando con los docentes.

Escritura de scripts (textos) y evaluación (paso 4). En esta fase, a cargo de los docentes, se escriben los textos por completo, tal como se filmarán, desglosando por unidad y videos de cada una. En nuestra experiencia, el uso de documentos en “Google Drive”, con un adecuado versionado, facilitó el proceso y el trabajo en simultáneo de varios docentes. En el caso de “¡A Programar!”/“Code Yourself!”, dado que ambas docentes a cargo eran bilingües, la escritura se hizo en el idioma materno de cada una para simplificar, combinando partes en inglés y partes en español. La versión final completa se tradujo a español e inglés. La escritura del texto permite también el uso posterior de “teleprompter” para la filmación (elemento que facilita dichas filmaciones), así como acelerar la tarea de subtítulo y generación de documentos pdf con el contenido textual de cada unidad. En el documento se incluyen las evaluaciones, cuyo diseño debe tener en cuenta las opciones que brinda la plataforma y sus limitaciones, además de las consideraciones académicas.

Revisión de scripts y definición de recursos (paso 5). Aquí se revisan a fondo los textos producidos, contando con el apoyo de expertos en comunicación y en

educación. Se realizan los ajustes necesarios. Se definen además qué elementos se destacarán con recursos audiovisuales, tales como definiciones, links, extractos de código o palabras clave, para luego, en la siguiente etapa definir su formato. La búsqueda de imágenes y/o sonidos de uso libre es delegado al equipo de producción, quien además guarda registro de las respectivas licencias para asignar los créditos correspondientes. Estos elementos se pueden ir subiendo a la plataforma al área de recursos correspondiente a la respectiva unidad (por ejemplo, si en un juego necesita una imagen de un personaje extraterrestre para la Unidad 4, ya se puede subir dicha imagen a la plataforma).

Preparación para la filmación (paso 6). Con el apoyo del productor audiovisual y de la dirección de arte, a partir de los recursos seleccionados, se detallan sus características concretas de la filmación: colores, diseño, formatos, etc. Por ejemplo, para el caso de App Inventor se decidió que todos los textos informativos, como el caso de un link, aparezcan con un ícono animado desde la derecha de la pantalla, con el personaje de Android deslizándose sobre un monopatín y desaparezca por la izquierda de la pantalla, dejando tras de sí el respectivo link fijo en la pantalla durante un cierto tiempo. Se definen además todos los otros recursos, tales como dibujos, animaciones y “stop motion”, manteniendo un mismo estilo, considerando el público objetivo. Se eligen también las locaciones de grabación. En “¡A Programar!”, se decidió que las partes en las cuales el docente explica un contenido teórico, sin uso de computadora, se filmaran en un lugar interesante, como por ejemplo una ventana de una oficina con linda vista panorámica, pero usándose una cortina para dejar translucir ese fondo sin que genere distracción. Para el caso de presentación de ejemplos en la computadora, se decidió usar un escritorio y el fondo con telón verde (para poder incluir imágenes y efectos posteriormente), con el docente sentado al lado de una laptop.

Como los cursos son de programación, hay que grabar tanto al docente como lo producido en la computadora y/o tableta. Para la filmación de “App Inventor”, que utiliza ambos elementos, fue necesario buscar, probar y seleccionar software que permita la grabación simultánea. En nuestro caso, la opción elegida para tal propósito fue Mobizen [32]. En la etapa de preparación de filmación también se definen los colores de ropa a utilizar, por ejemplo, en el caso de dos docentes juntos se indican colores y estilos que combinen. En el primero de los MOOCs, los videos se filmaron sin la ayuda de “teleprompter” (por no disponerlo en ese momento), lo cual exigía un alto grado de preparación de los docentes para la filmación: la explicación del desarrollo de programas lleva un orden bastante estricto y omitir un cierto paso implicaba que se debía volver a filmar esa parte. En los últimos MOOCs se ha incluido el uso de este dispositivo utilizando como insumo los textos de los docentes, lo cual simplifica la tarea de filmación para el docente. En esta etapa, se prepara el texto en el formato requerido para “teleprompter”.

Filmación de una unidad de prueba (paso 7). Aquí participan tanto el equipo docente como el equipo de producción. En nuestra experiencia conviene grabar por completo una unidad que tenga todo tipo de recursos y que no sea solamente exposición oral del docente, para detectar posibles ajustes a realizar. Se ensaya con

los docentes sobre el uso del “teleprompter”. Se tienen en cuenta también los lugares elegidos para filmación. En el caso de “¡A Programar!”, se realizaron varias pruebas para elegir la más apropiada (distintas ubicaciones con fondos diferentes, incluyendo día soleado y día lluvioso). Con el fin de lucir más natural en la filmación y en la medida de lo posible, los videos se pueden grabar en otro orden, dejando para el final el video introductorio de presentación de la unidad, pues en general luego de filmar algunos el proceso fluye mejor. Se afinan detalles del maquillaje y se hacen también pruebas de cámara con la ropa que se utilizará en las demás unidades. Por ejemplo, en “Scratch 2.0 Avanzado” hemos elegido que cada unidad tenga un color que la represente como refuerzo visual de la unidad, los docentes usan ropas en esos tonos y se incluyó en los videos un detalle de ese color, así como en los documentos pdf que contienen el texto de cada video.

Revisión y ajustes de la unidad de prueba (paso 8). En esta etapa, se realizan todos los procesos de postproducción y se trata de dejar ya finalizada y pronta esta unidad. Se detectan todos los detalles a mejorar, filmándose nuevamente lo que sea necesario o ajustando la edición para lograr los efectos deseados. Para todo el proceso de revisión es recomendable que participen tanto los técnicos de producción audiovisual, así como los propios docentes, pues cada uno percibe detalles diferentes. También se incluyen pruebas con el público objetivo. Por ejemplo en “Code Yourself!” se mostraron algunos de los primeros videos a diferentes posibles alumnos para obtener sus comentarios en términos de estilo, contenido, ritmo, etc. A través de una planilla compartida, se registran todas las oportunidades de mejora de los videos y se analiza la viabilidad de cada una, por ejemplo, incluir o no otra animación. De esta manera, se establecen bases firmes para la filmación y proceso de las demás unidades.

Filmación y revisión de las demás unidades del curso (paso 9). Con las bases establecidas, se filman y revisan las demás unidades del curso. En base a la experiencia realizada en la unidad filmada, el proceso se realiza más rápidamente y sin dificultades. También aquí se revisan muy cuidadosamente todos los videos, realizándose un seguimiento detallado de las versiones de edición. Cada detalle a ajustar o sugerencia es registrada, indicándose el minuto y segundo correspondiente del respectivo video. Cada una de ellas es evaluada por el coordinador de producción en conjunto con el docente responsable del curso y se decide si se realiza o no. Contar con los textos escritos es de gran ayuda para la edición y revisión. Es de destacar que en todas las etapas del proceso, siempre se están buscando oportunidades de mejora. Por ejemplo, durante la preparación de App Inventor, en primera instancia se subían al área compartida de trabajo los videos en baja definición, para minimizar tiempos de subida. Esto implicaba que, en el caso de ya ser las versiones finales había que volver a subirlos en alta definición y revisar nuevamente. Este proceso consumía excesivo tiempo de los docentes y demás integrantes del equipo de producción que revisaban los videos. Se optó por subir siempre videos en alta definición, aún en etapas preliminares.

Subida a la plataforma (paso 10). A medida que los videos están prontos y completamente revisados, se suben a la plataforma. Los cuestionarios y demás

evaluaciones se van subiendo desde que están verificados. Se establecen además los requisitos de “precedencia”, por ejemplo, en la plataforma CREA2 se decidió poner como requisito para ver un video de una cierta unidad, el haber visto los videos de las unidades previas. En el caso de Coursera, están todos los materiales disponibles desde el comienzo. También se crean los foros necesarios y se indican las características de aprobación del curso en la propia plataforma. Además, se incluye en cada curso información detallada sobre fechas y aprobación, así como una descripción de los diversos materiales, para simplificar el acceso de los participantes.

Pretesteo (paso 11). En la medida de lo posible, se realiza una simulación completa con algunos alumnos en la propia plataforma para probar todos los elementos del curso. Se ajustan los posibles detalles que aparezcan. También aquí es recomendable llevar claro registro de los errores u oportunidades de mejora, así como las acciones correspondientes.

Lanzamiento y seguimiento (paso 12). A partir de esta etapa, se pone disponible el curso a los participantes. Durante la primera sesión se realiza un seguimiento muy detallado, tratando de detectar cualquier posible dificultad para ajustarla lo antes posible. En el caso de Coursera, donde “¡A Programar!” y “Code Yourself!” están disponibles desde hace más de un año, se sigue realizando monitoreo por parte de docentes de apoyo. En el caso de Plan Ceibal, el seguimiento de las posteriores sesiones es a cargo de su propio personal.

Es importante destacar que si bien podrían considerarse algunas etapas como eventualmente más relevantes, como por ejemplo escribir el script por completo y verificarlo, todas y cada una de ellas son fundamentales para lograr un buen producto final. Esta misma consideración se presenta en cuanto a los integrantes de los diferentes equipos. El coordinador general desempeña un papel primordial, integrando los distintos componentes, monitoreando y ajustando todo el proceso, pero cada uno de los demás participantes también tienen un papel clave en el desarrollo global.

5 Conclusiones

En este trabajo se presentaron los MOOCs realizados por la Universidad ORT Uruguay para Plan Ceibal y el realizado en conjunto con The University of Edinburgh, todos ellos dirigidos a la enseñanza inicial de programación para jóvenes. Estos MOOCs tienen por objetivo contribuir al desarrollo de las capacidades de pensamiento lógico y computacional en los jóvenes, habilidades cada vez más demandadas en la sociedad actual. Los lenguajes utilizados son Scratch y MIT App Inventor. Las encuestas realizadas a participantes de los MOOCs disponibles muestran un alto grado de conformidad.

A partir de las distintas experiencias se diseñó una propuesta de modelo para el proceso de diseño de MOOCs para aprender a programar, que puede ser utilizado como guía para el desarrollo y que, además, podría ser extrapolado a otros dominios o áreas de conocimiento. El modelo consta de 12 pasos que van desde la conformación de los equipos de trabajo hasta el lanzamiento y seguimiento del curso. En las

distintas etapas se incluyen recomendaciones y lecciones aprendidas, así como se detallan los distintos participantes. Como trabajo futuro está previsto ampliar este modelo para el caso de traducción de un curso ya completo a otra lengua, lo que implica definir etapas para el proceso de migración y adaptación. En particular, se está trabajando con la Universidad de San Pablo para lanzar una versión de “¡A Programar!” en portugués (“Programe-se”).

Referencias

1. ACM Computer Science Teachers Association. CSTA K–12 Computer Science Standards - 2011), <http://csta.acm.org/Curriculum/sub/K12Standards.html>
2. Balanskat, A., Engelhardt, K. (contributors): “Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europa”. European Schoolnet, Belgium. (Oct 2015)
3. Department for Education, “National curriculum in England: computing programmes of study (2013)”, <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>
4. Cooper, S., Sahami, M.: “Reflections on Stanford’s MOOCs”, Communications of ACM, V 56, N 2. (Feb 2013)
5. Universities UK: “Massive open online courses. Higher education’s digital moment?”, <http://www.universitiesuk.ac.uk/highereducation/Documents/2013/MassiveOpenOnlineCourses.pdf>
6. Oxford Dictionary, http://www.oxforddictionaries.com/us/definition/american_english/MOOC
7. Siever, W.: “Leveraging MOOCs”, JCSC 30, 2 (Dec 2014)
8. Mc Auley, A., Stewart, B., Siemens, G., Cormier, D. : “The MOOC Model for Digital Practice”. http://davecormier.com/edblog/wp-content/uploads/MOOC_Final.pdf
9. Edinburgh MOOCs handbook: “How to grow your own MOOC with Coursera”. http://moocs.unige.ch/files/3414/2503/3908/Growing_an_Edinburgh_MOOC.pdf
10. Northern Illinois University: “Tips for Designing a Massive Open Online Course (MOOC)”, <http://facdevblog.niu.edu/tips-for-designing-a-massive-open-online-course-mooc>
11. Yousef, A.M.F., Chatti, M. A., Schroeder, U., Wosnitza, M.: "What Drives a Successful MOOC? An Empirical Examination of Criteria to Assure Design Quality of MOOCs", Conf. on Advanced Learning Technologies (ICALT), Greece (2014)
12. Coursera, <https://www.coursera.org/>
13. Edsurge: “Udacity, Coursera and edX Now Claim Over 24 Million Students“, <https://www.edsurge.com/news/2015-09-08-udacity-coursera-and-edx-now-claim-over-24-million-students>
14. Spyropoulou, N., Demopoulou, G., Pierrakeas, C, Koutsoukos, I., Kameas, A.: “Developing a Computer Programming MOOC”. Procedia Computer Science, 65, 182-191 (2015)
15. Collazos, A.: “8 pasos para crear un MOOC”, Revista de Educación Virtual. <http://revistaeducacionvirtual.com/archives/1379>
16. Alario Hoyos, C., Pérez-Sanagustín, M., Delgado, C., Muñoz-Merino, P.J.: "Recommendations for the design and deployment of MOOCs: insights about the MOOC digital education of the future deployed in MiríadaX", TEEM '14 Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality, Pp 403-408, ACM New York, NY, USA (2014)

17. Morrison, M.: "Teaching Tips From a Master MOOC-Maker", <https://onlinelearninginsights.wordpress.com/2013/04/15/teaching-tips-from-a-master-mooc-maker/>
18. Bali, M. : "MOOC Pedagogy: Gleaning Good Practice from existing MOOCs", MERLOT J. of online Learning and Teaching, V. 10, N.1 (Mar 2014)
19. Button, K.: "10 Lessons learned from Moocs", Education Dive, <http://www.educationdive.com/news/10-lessons-learned-from-moocs/306113/>
20. Guo, P. J., Kim, J., Rubin, R. : "How video production affects student engagement: an empirical study of MOOC videos", L@S '14 Proceedings of the first ACM conference on Learning @ scale conference, USA (2014)
21. Hibbert, M.: "What Makes an Online Instructional Video Compelling?" <http://er.educause.edu/articles/2014/4/what-makes-an-online-instructional-video-compelling>
22. Cornell University: "Tips on creating video for your course!", <http://acadtech.cit.cornell.edu/2014/04/28/tips-on-creating-video-for-your-course/>
23. Adamopoulos, P.: "What makes a great MOOC? An interdisciplinary analysis of student retention in online courses". 34th International Conference on Information Systems, Milan (2013)
24. Kereki, I., Paulós, J.V. : "SM4T: Scratch MOOC for Teens - A pioneer pilot experience in Uruguay", FIE 2014 Frontiers in Education, Spain (2014)
25. Plan Ceibal, <http://www.ceibal.edu.uy>
26. Universidad ORT Uruguay y The University of Edinburgh: "MOOC: ¡A Programar!". <https://www.coursera.org/learn/a-programar>
27. The University of Edinburgh and Universidad ORT Uruguay."MOOC: Code Yourself!" <https://www.coursera.org/learn/intro-programming>
28. Scratch, <http://wiki.scratch.mit.edu/wiki/Scratch>
29. MIT AppInventor, <http://appinventor.mit.edu/>
30. Educativa: <http://www.educativa.com/>
31. Schoology: <https://www.schoology.com/>
32. Mobizen: <https://www.mobizen.com/>